Controlling Deployments using Release Gates

Overview

As you may be aware, a release pipeline specifies the end-to-end release process for an application to be deployed across a range of environments. Deployments to each environment are fully automated by using phases and tasks. Ideally, you do not want new updates to the applications to be exposed to all the users at the same time. It is a best practice to expose updates in a phased manner i.e. expose to a subset of users, monitor their usage and expose to other users based on the experience the initial set of users had.

Approvals and gates enable you to take control over the start and completion of the deployments in a release. With approvals, you can wait for users to manually approve or reject deployments. Using release gates, you can specify application health criteria that must be met before release is promoted to the next environment. Prior to or after any environment deployment, all the specified gates are automatically evaluated until they all pass or until they reach your defined timeout period and fail.

Gates can be added to an environment in the release definition from the pre-deployment conditions or the post-deployment conditions panel. Multiple gates can be added to the environment conditions to ensure all the inputs are successful for the release.

As an example:

Pre-deployment gates ensures there are no active issues in the work item or problem management system before deploying a build to an environment.

Post-deployment gates ensures there are no incidents from the monitoring or incident management system for the app after it's been deployed, before promoting the release to the next environment.

4 types of gates are included by default for every account.

1. **Invoke Azure function:** Trigger execution of an Azure function and ensures a successful completion. For more details, see Azure function task
2. **Query Azure monitor alerts:** Observe the configured Azure monitor alert rules for active alerts. For more details, see Azure monitor task
3. **Invoke REST API:** Make a call to a REST API and continue if it returns a successful response. For more details, see HTTP REST API task
4. **Query Workitems:** Ensures the number of matching work items returned from a query is within a threshold. For more details, see Query Workitems task

What's covered in this lab?

This lab covers the configuration of the deployment gates and details how to add the control to Azure pipelines. You will configure a release definition with two environments for an Azure Web App. You will deploy to the **Canary** environment only when there are no blocking bugs for the app and mark the Canary environment complete only when there are no active alerts in Azure Monitor (Application Insights).

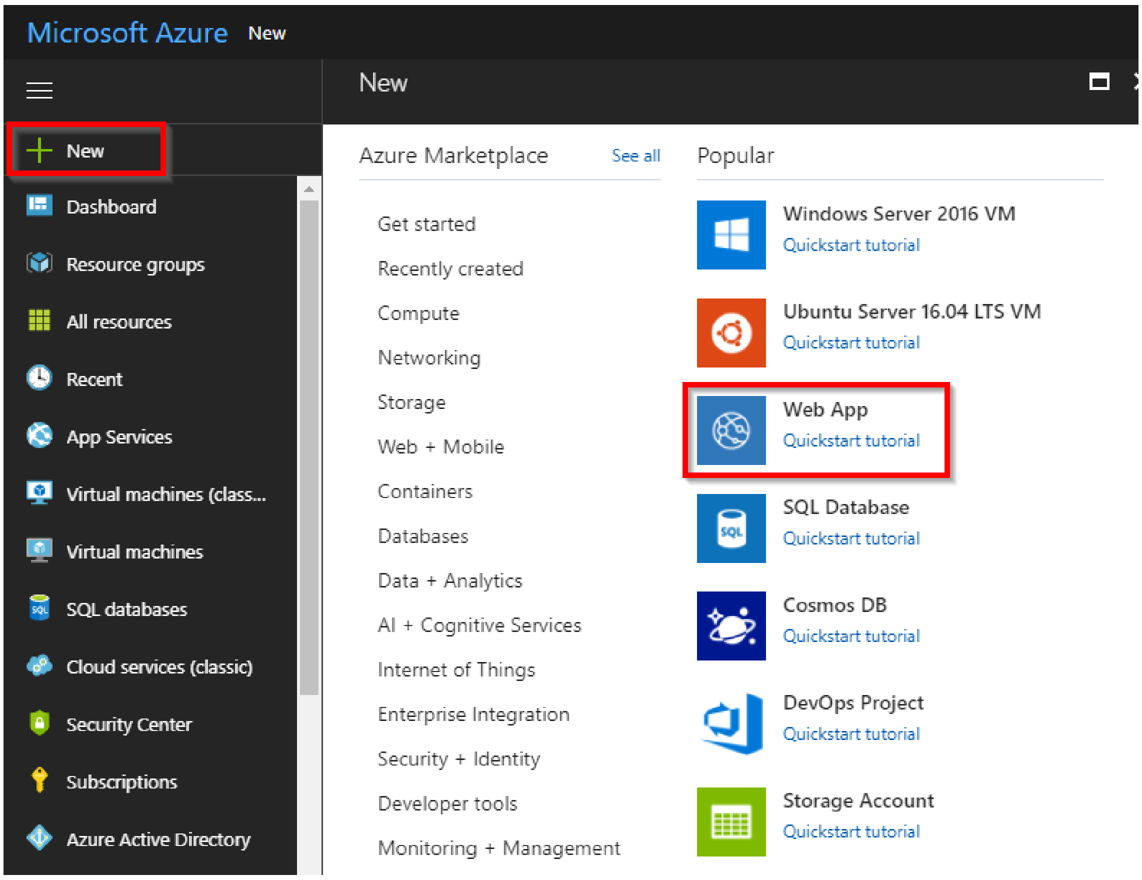
Before you begin

1. Refer the Getting Started page to know the prerequisites for this lab.
2. Click the Azure DevOps Demo Generator link and follow the instructions in Getting Started page to provision the project to your **Azure DevOps**.

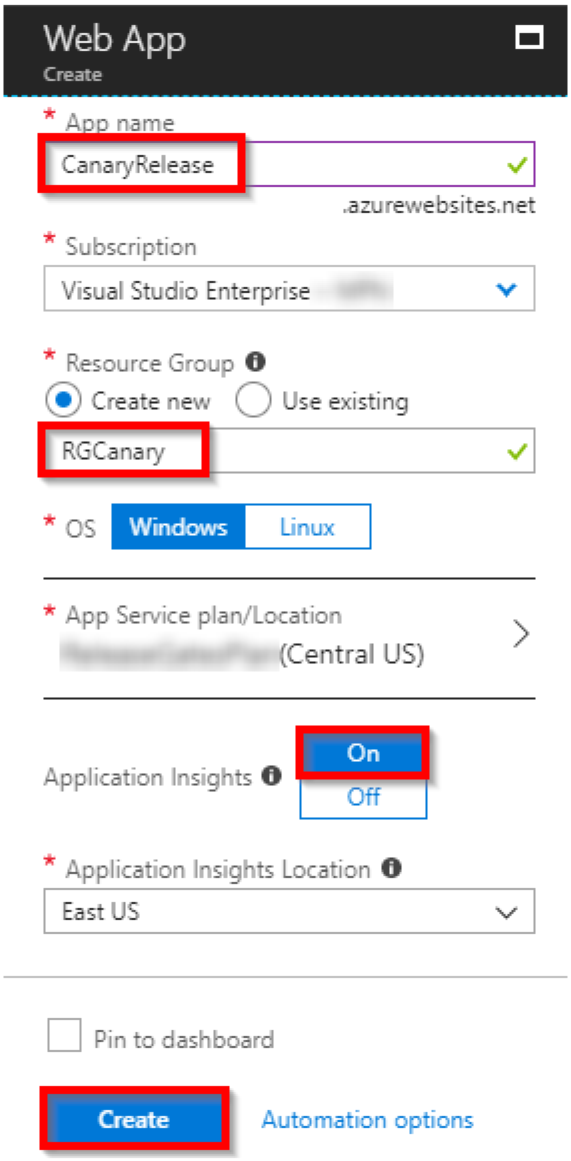
Setting up the Target Environment

You will create two **Web Apps** in Azure to depict two environments **Canary** and **Production** to deploy the application.

1. Go to Azure portal and click on **+New** and select **Web App**.



2. Provide a name for the **Web App**, create new **Resource Group** or select existing one from the dropdown. Enable **Application Insights** and click **Create**.



Web App
Create

* App name
CanaryRelease
.azurewebsites.net

* Subscription
Visual Studio Enterprise [redacted]

* Resource Group
[✓ Create new] Use existing
RGCanary

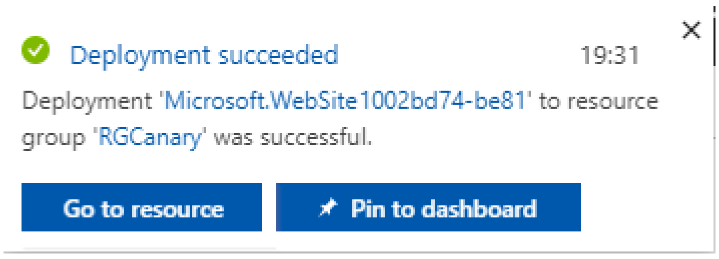
* OS [✓ Windows] Linux

* App Service plan/Location
[redacted](Central US)

Application Insights [✓ On] Off

* Application Insights Location
East US

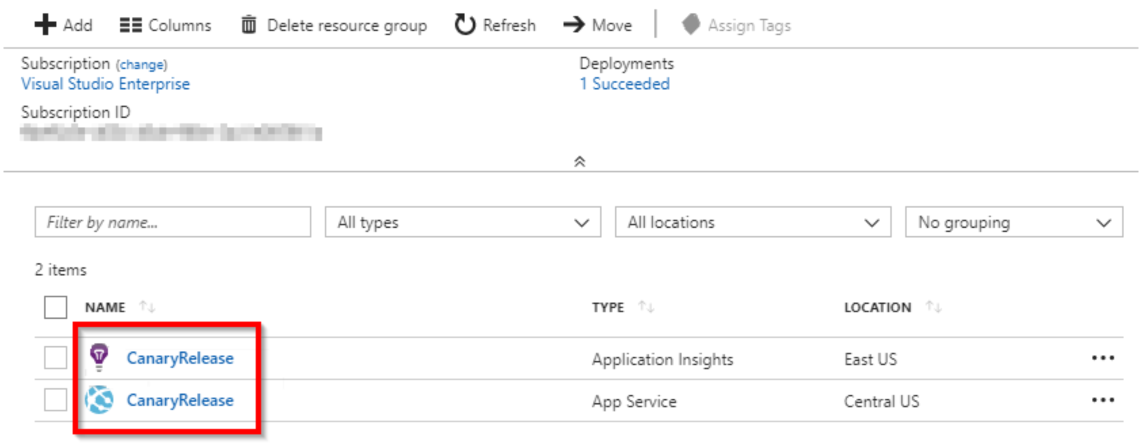
3. Once the deployment succeeds, navigate to your **Resource Group** to see the resources created.



Deployment succeeded 19:31

Deployment 'Microsoft.WebSite1002bd74-be81' to resource group 'RGCanary' was successful.

4. You will see a Web App and an Application Insights being provisioned. Application Insights is used to monitor the Web app.



Subscription (change)
Visual Studio Enterprise
Subscription ID
[redacted]
Deployments
1 Succeeded

Filter by name... | All types | All locations | No grouping

2 items

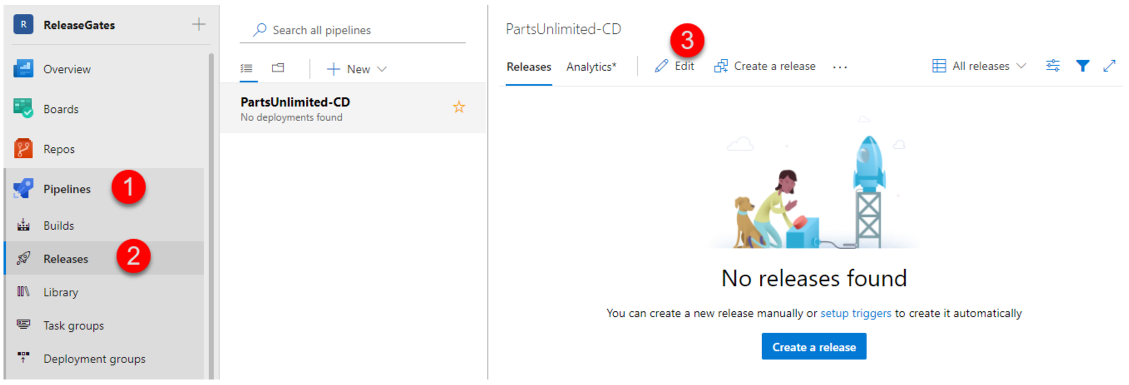
NAME	TYPE	LOCATION
CanaryRelease	Application Insights	East US
CanaryRelease	App Service	Central US

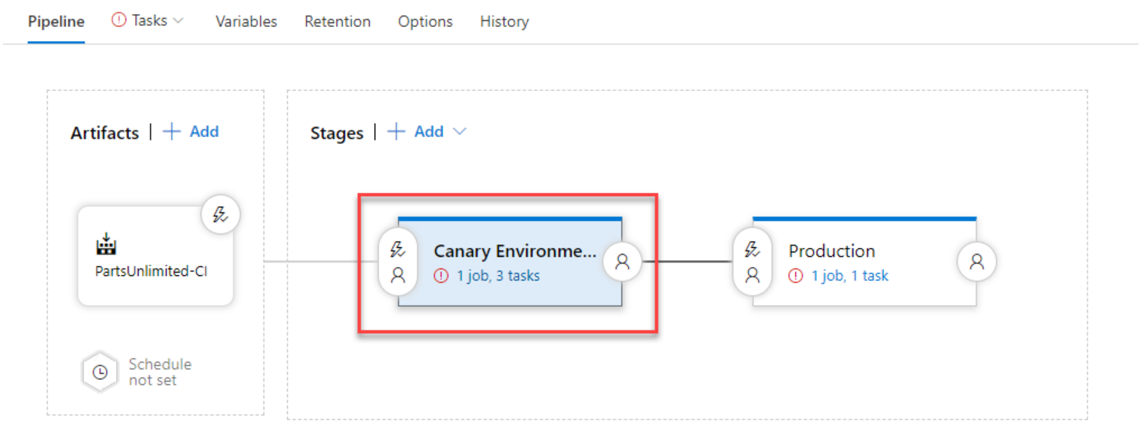
5. Repeat **Step 1 & Step 2** to create Web App for **Production**.

Exercise 1: Configure Release pipeline

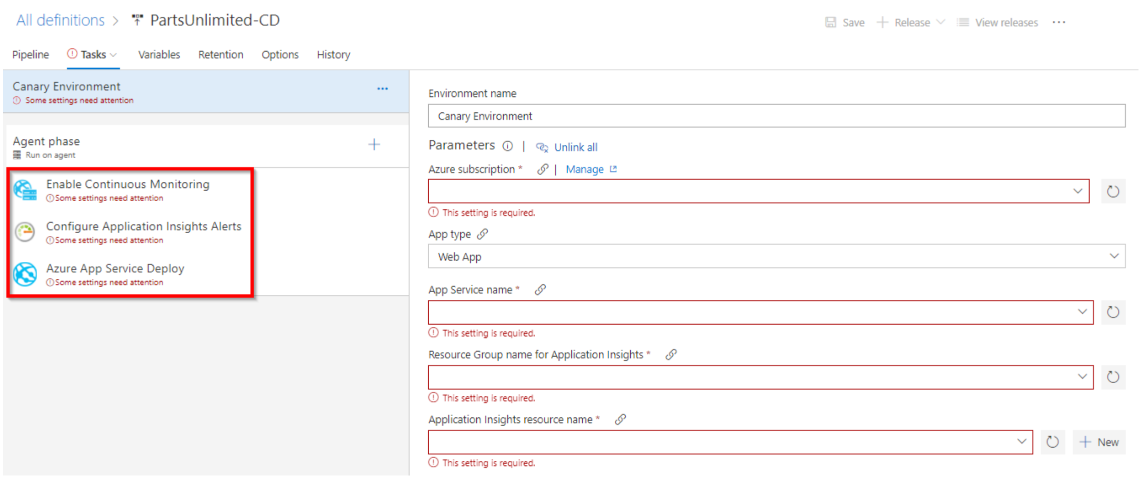
Update Release Tasks

1. Navigate to **Releases** under **Pipelines** and **Edit** the pipeline **PartsUnlimited-CD**. In this pipeline, you have two environments viz. *Canary Environment* & *Production*. Click on **"1 job, 3 tasks"** link for Canary Environment to update the tasks.





2. The canary environment has 3 tasks which will publish the package to Azure Web App, enables continuous monitoring of the application after deployment and also Application Insights Alerts will be configured.


All definitions > PartsUnlimited-CD

Save | Release | View releases | ...

Pipeline | Tasks | Variables | Retention | Options | History

Canary Environment
Some settings need attention

Agent phase
Run on agent

- Enable Continuous Monitoring
  Some settings need attention
- Configure Application Insights Alerts
  Some settings need attention
- Azure App Service Deploy
  Some settings need attention

Environment name: Canary Environment

Parameters | Unlink all

Azure subscription * | Manage
This setting is required.

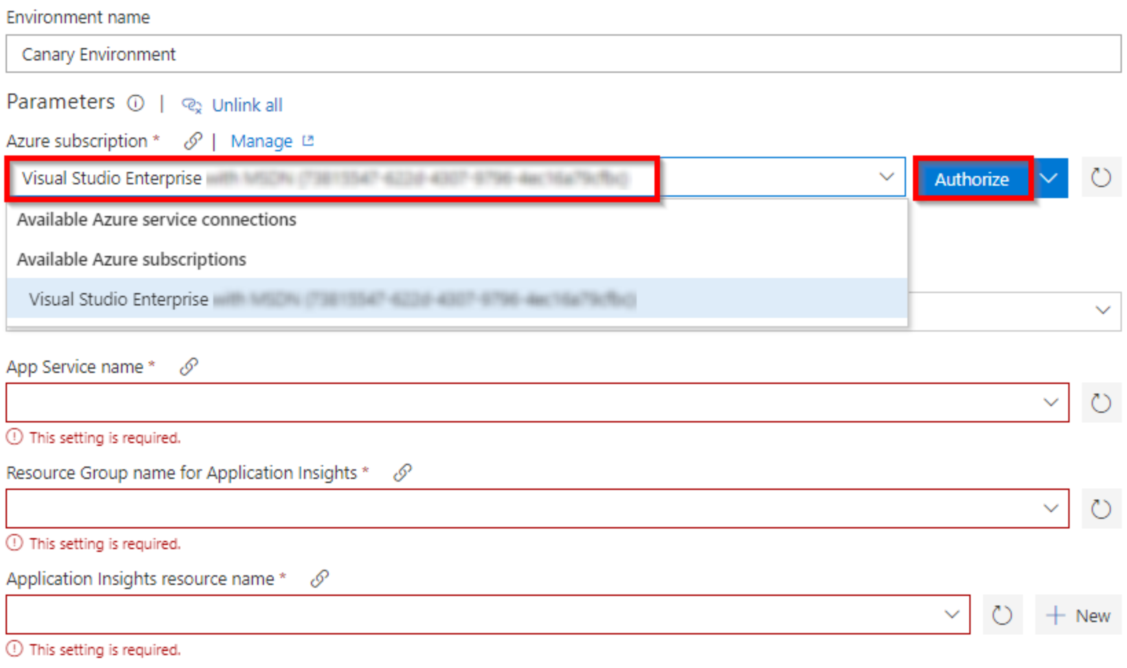
App type: Web App

App Service name *
This setting is required.

Resource Group name for Application Insights *
This setting is required.

Application Insights resource name * | New
This setting is required.

3. In Azure Subscription field, select your Azure subscription from the dropdown and click on **Authorize**. Provide your credentials, if required, to complete the authorization to your Azure account.


Environment name: Canary Environment

Parameters | Unlink all

Azure subscription * | Manage

Visual Studio Enterprise [redacted]

Authorize

Available Azure service connections

Available Azure subscriptions

Visual Studio Enterprise [redacted]

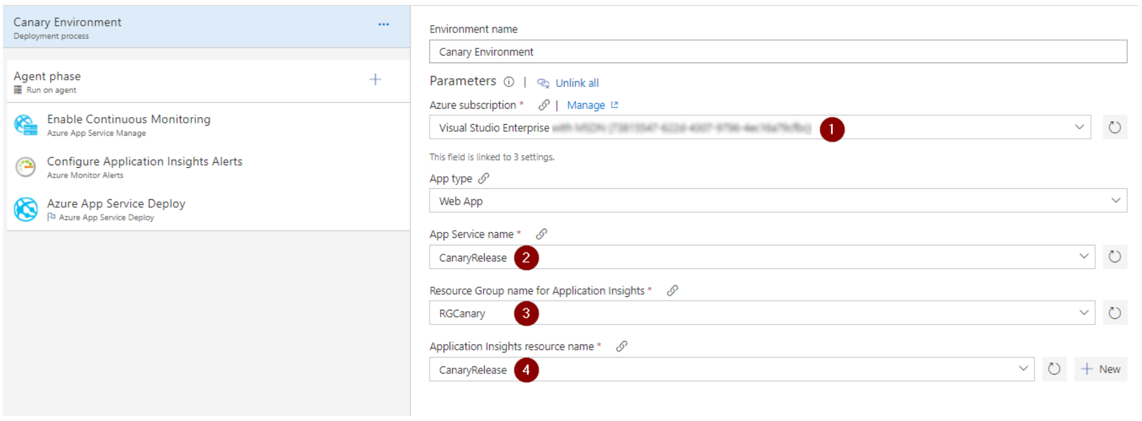
App Service name *
This setting is required.

Resource Group name for Application Insights *
This setting is required.

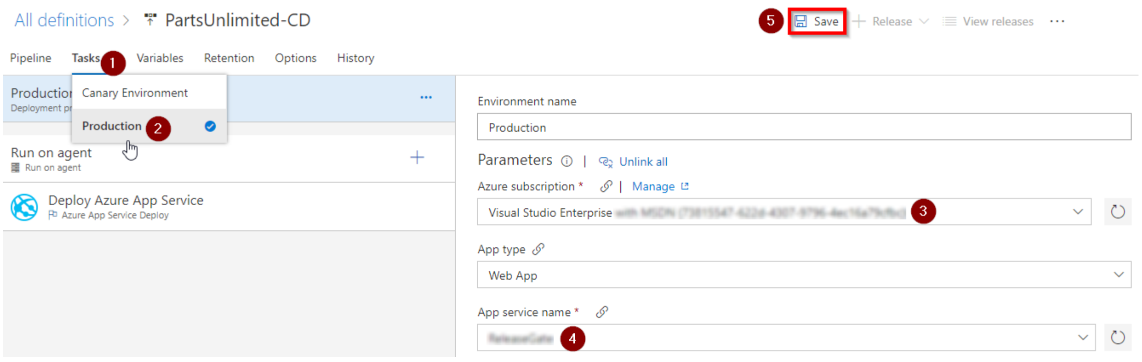
Application Insights resource name * | New
This setting is required.

Note: Disable pop-up blocker in your browser if you see a blank screen after clicking Authorize, and retry the step.

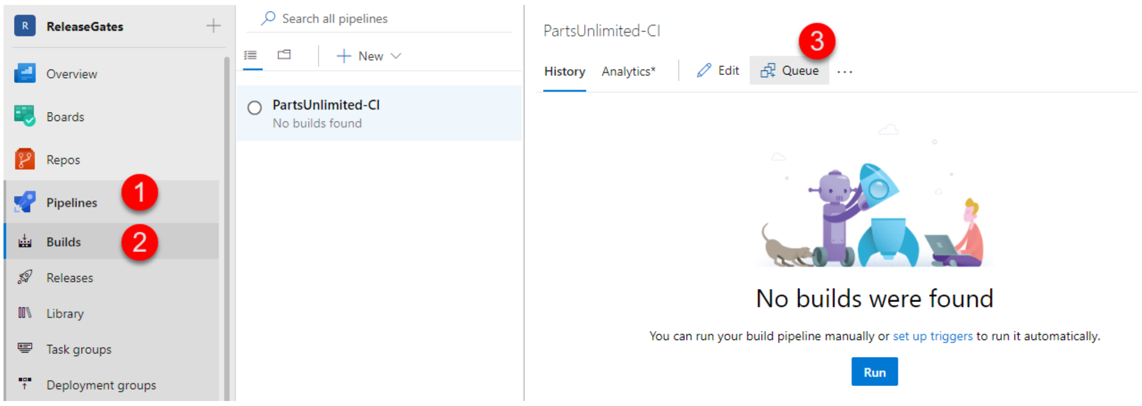
4. Select the App Service, Resource Group and Application Insights that you created for Canary environment from the drop-down.

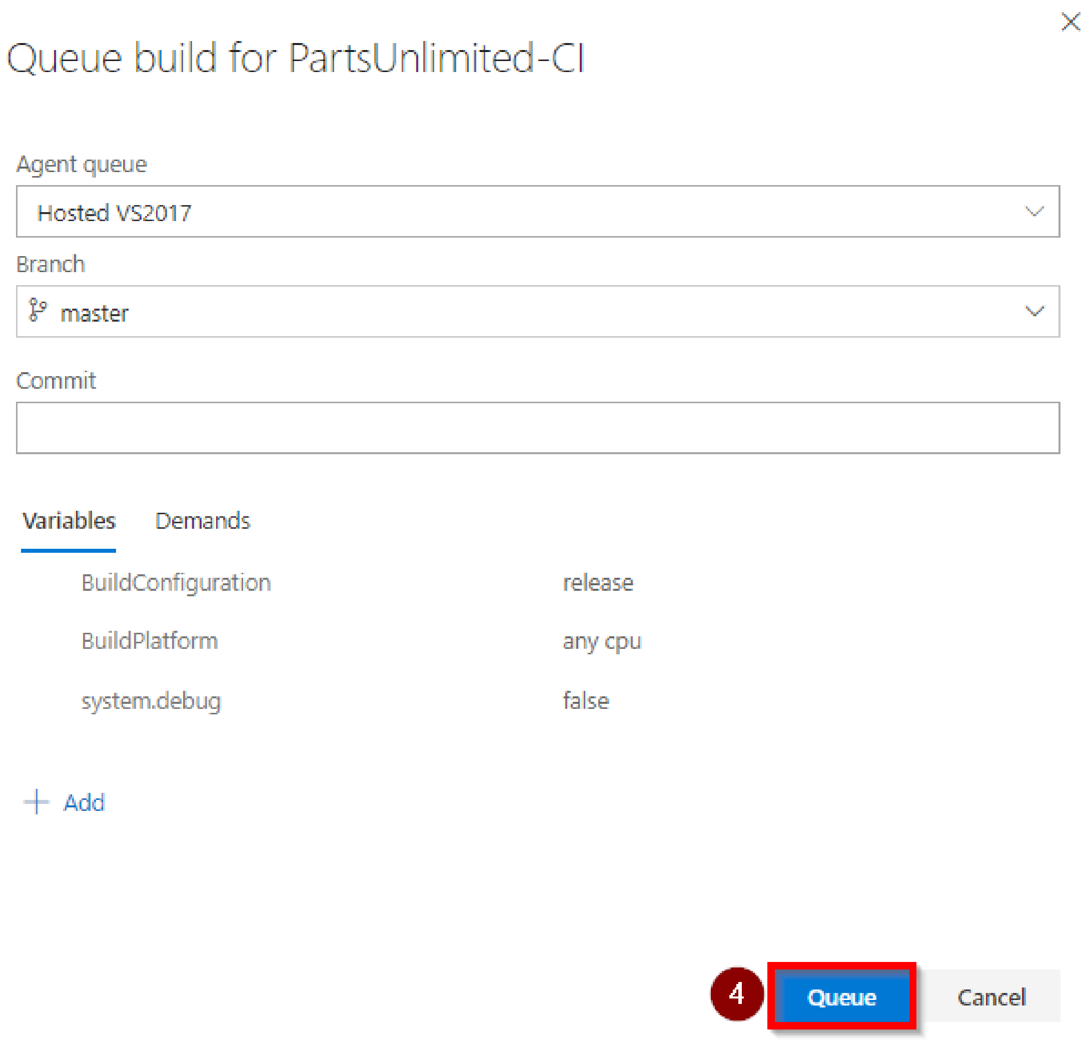


5. For Production, select the **Azure subscription** from the drop-down. Pick the App service you created for Production and click on **Save** button.



6. Navigate to **Builds** under **Pipelines** and **Queue new build** for the build definition **PartsUnlimited-CI**.




Queue build for PartsUnlimited-CI

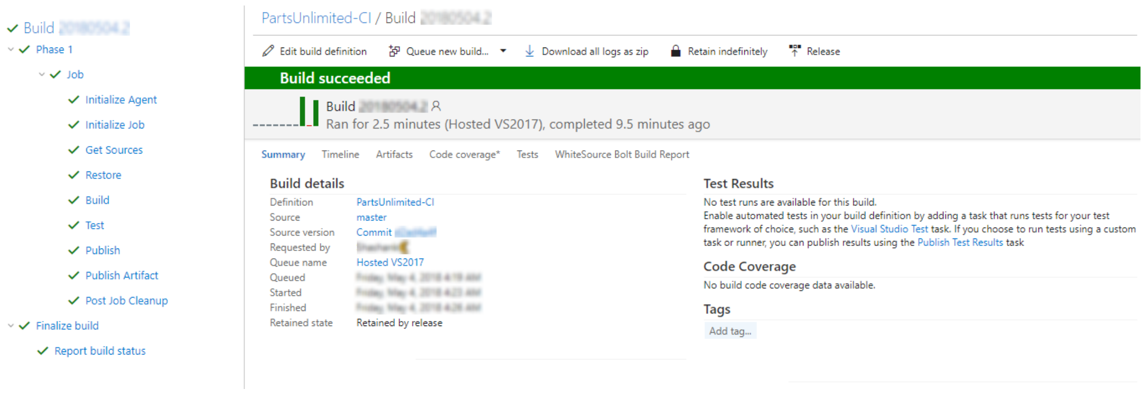
Agent queue: Hosted VS2017

Branch: master

Commit:

BuildConfiguration	release
BuildPlatform	any cpu
system.debug	false

7. After the build succeeds, the release will be triggered automatically and the application will be deployed to both the environments. Browse the websites after the application is deployed.



Build succeeded

Build [redacted]
Ran for 2.5 minutes (Hosted VS2017), completed 9.5 minutes ago

Build details

Definition	PartsUnlimited-CI
Source	master
Source version	Commit [redacted]
Requested by	[redacted]
Queue name	Hosted VS2017
Queued	[redacted]
Started	[redacted]
Finished	[redacted]
Retained state	Retained by release

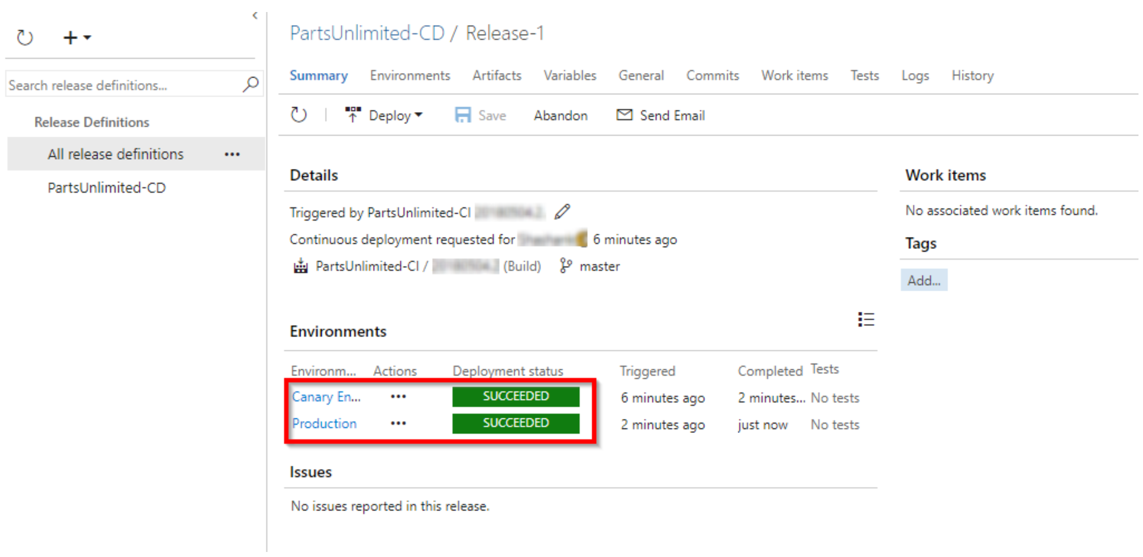
Test Results

No test runs are available for this build.
Enable automated tests in your build definition by adding a task that runs tests for your test framework of choice, such as the Visual Studio Test task. If you choose to run tests using a custom task or runner, you can publish results using the Publish Test Results task

Code Coverage

No build code coverage data available.

Tags


Search release definitions...

Release Definitions

All release definitions

PartsUnlimited-CD

PartsUnlimited-CD / Release-1

Summary | Environments | Artifacts | Variables | General | Commits | Work items | Tests | Logs | History

Deploy | Save | Abandon | Send Email

Details

Triggered by PartsUnlimited-CI [redacted]

Continuous deployment requested for [redacted] 6 minutes ago

PartsUnlimited-CI / [redacted] (Build) master

Work items

No associated work items found.

Tags

Add...

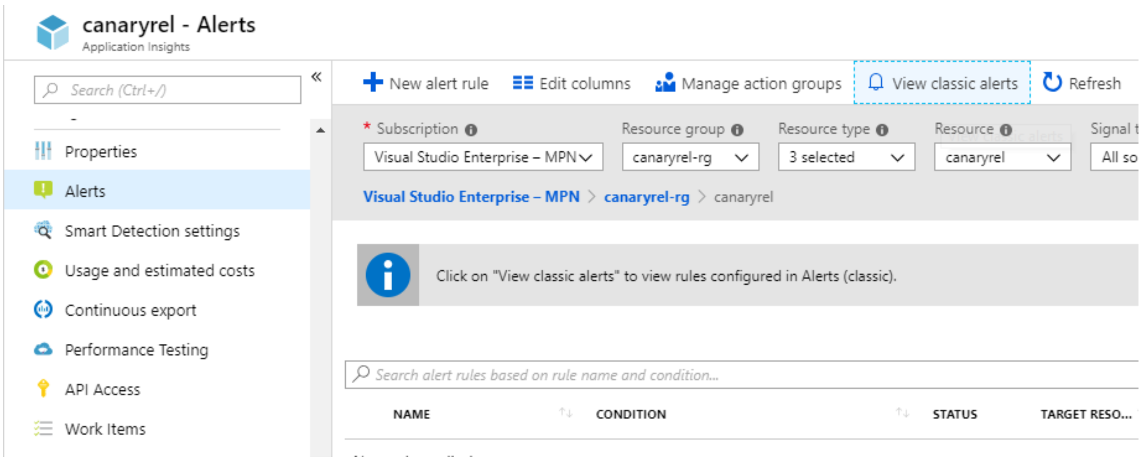
Environments

Environm...	Actions	Deployment status	Triggered	Completed	Tests
Canary En...	•••	SUCCEEDED	6 minutes ago	2 minutes...	No tests
Production	•••	SUCCEEDED	2 minutes ago	just now	No tests

Issues

No issues reported in this release.

8. This will automatically hook the Web App with Application Insights and configure the Alerts in Azure under *CanaryRelease* Application Insights **Alerts** section. Click **View classic alerts** to view the metrics.


canaryrel - Alerts
Application Insights

Search (Ctrl+/)

Properties
Alerts
Smart Detection settings
Usage and estimated costs
Continuous export
Performance Testing
API Access
Work Items

New alert rule | Edit columns | Manage action groups | View classic alerts | Refresh

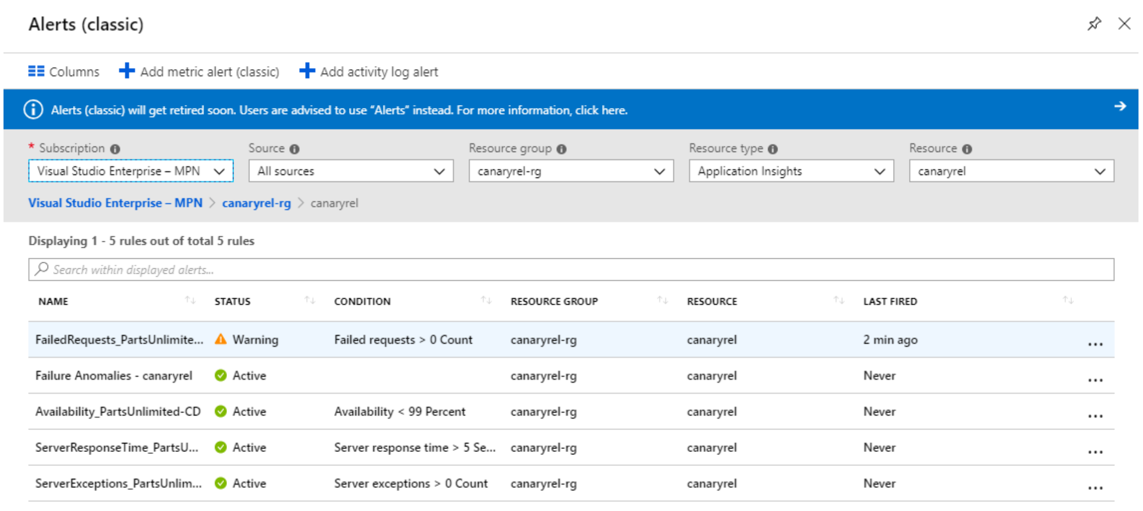
* Subscription	Resource group	Resource type	Resource	Signal t
Visual Studio Enterprise – MPN	canaryrel-rg	3 selected	canaryrel	All so

Visual Studio Enterprise – MPN > canaryrel-rg > canaryrel

Click on "View classic alerts" to view rules configured in Alerts (classic).

Search alert rules based on rule name and condition...

NAME	CONDITION	STATUS	TARGET RESO...


Alerts (classic)

Columns | Add metric alert (classic) | Add activity log alert

Alerts (classic) will get retired soon. Users are advised to use "Alerts" instead. For more information, click here.

* Subscription	Source	Resource group	Resource type	Resource
Visual Studio Enterprise – MPN	All sources	canaryrel-rg	Application Insights	canaryrel

Visual Studio Enterprise – MPN > canaryrel-rg > canaryrel

Displaying 1 - 5 rules out of total 5 rules

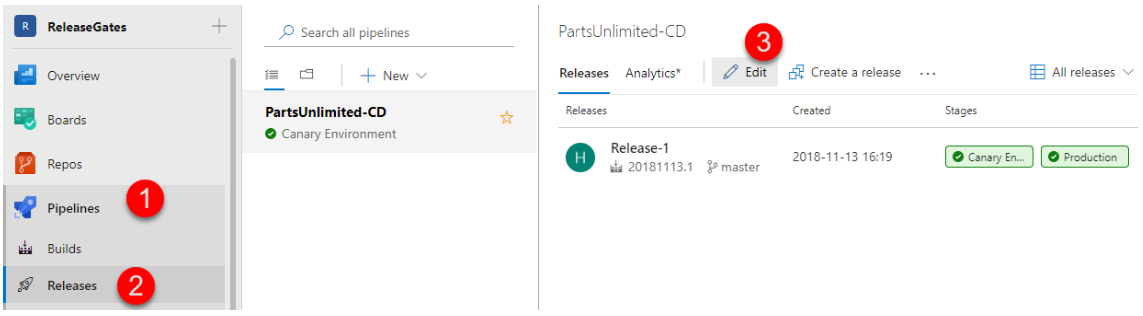
Search within displayed alerts...

NAME	STATUS	CONDITION	RESOURCE GROUP	RESOURCE	LAST FIRED	
FailedRequests_PartsUnlimite...	⚠ Warning	Failed requests > 0 Count	canaryrel-rg	canaryrel	2 min ago	...
Failure Anomalies - canaryrel	✓ Active		canaryrel-rg	canaryrel	Never	...
Availability_PartsUnlimited-CD	✓ Active	Availability < 99 Percent	canaryrel-rg	canaryrel	Never	...
ServerResponseTime_PartsU...	✓ Active	Server response time > 5 Se...	canaryrel-rg	canaryrel	Never	...
ServerExceptions_PartsUnlim...	✓ Active	Server exceptions > 0 Count	canaryrel-rg	canaryrel	Never	...

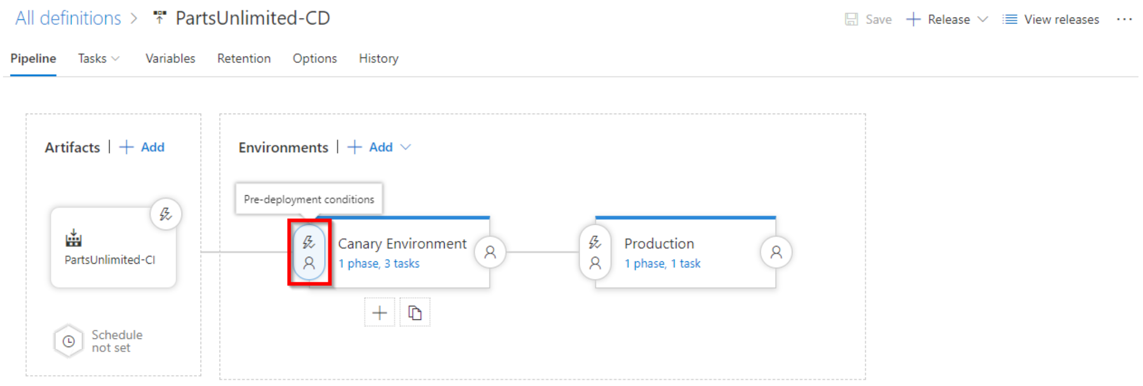
Exercise 2: Configure Deployment Gates.

Enabling Pre-deployment Gate

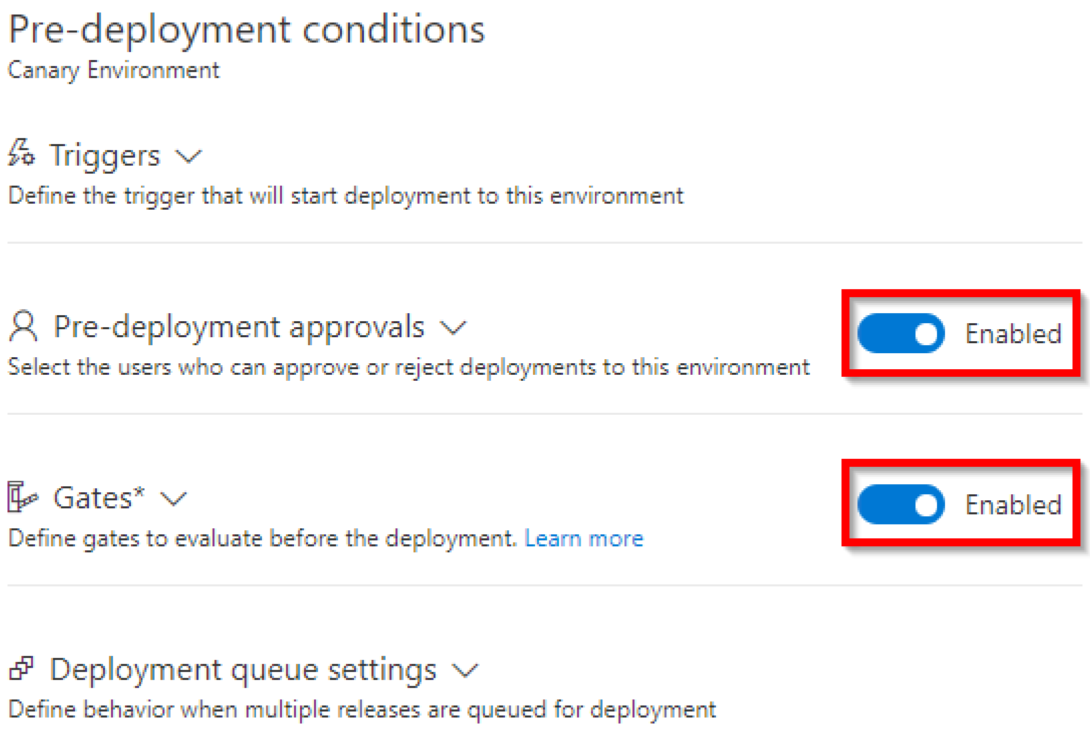
1. Edit the release pipeline **PartsUnlimited-CD** in *Releases* under **Pipelines**.



2. Click on **Pre-deployment conditions**.



3. You will see **Triggers**, **Pre-deployment approvals**, **Gates** and **Deployment queue settings**. Enable **Pre-deployment approvals** and **Gates**.



Pre-deployment conditions
Canary Environment

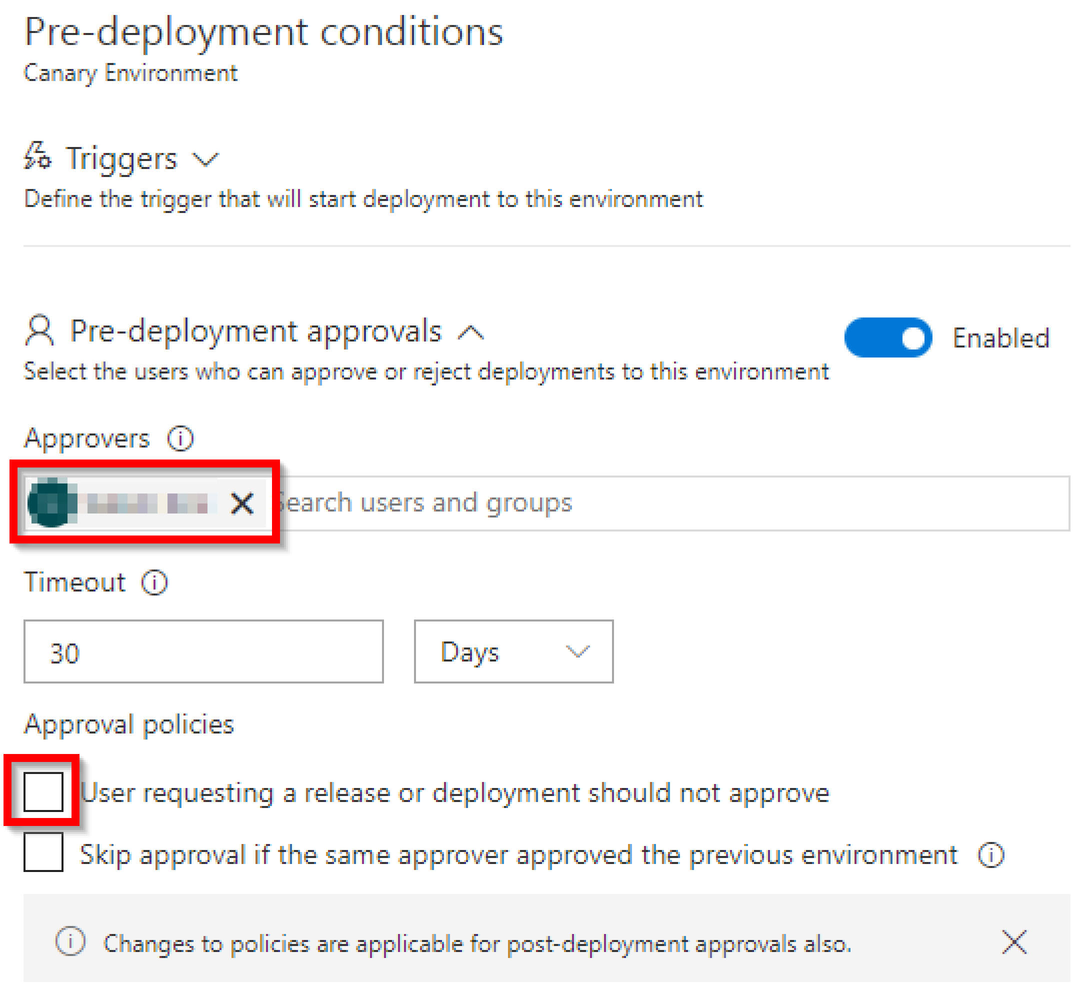
Triggers
Define the trigger that will start deployment to this environment

Pre-deployment approvals — Enabled
Select the users who can approve or reject deployments to this environment

Gates* — Enabled
Define gates to evaluate before the deployment. Learn more

Deployment queue settings
Define behavior when multiple releases are queued for deployment

4. Add yourself as an **Approver** and by default, the user requesting a release or deployment should not approve. However, for the purpose of this lab, **uncheck** this condition.


Pre-deployment conditions
Canary Environment

Triggers
Define the trigger that will start deployment to this environment

Pre-deployment approvals
Select the users who can approve or reject deployments to this environment
Enabled

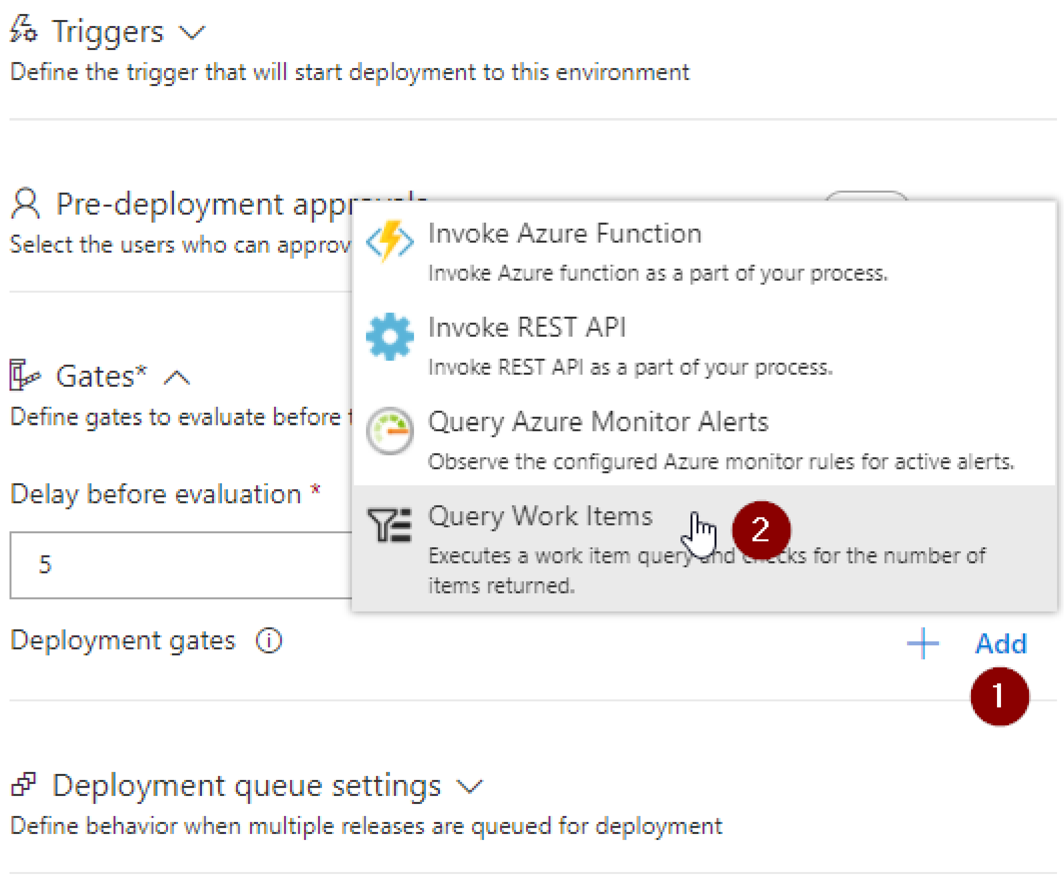
Approvers
[redacted] Search users and groups

Timeout
30 Days

Approval policies
☐ User requesting a release or deployment should not approve
☐ Skip approval if the same approver approved the previous environment

Changes to policies are applicable for post-deployment approvals also.

5. Add **Query Work Items** to the Gates.


Triggers

Define the trigger that will start deployment to this environment

Pre-deployment approvals

Select the users who can approve

Gates*

Define gates to evaluate before

Delay before evaluation *

5

Deployment gates

Add

1

Invoke Azure Function

Invoke Azure function as a part of your process.

Invoke REST API

Invoke REST API as a part of your process.

Query Azure Monitor Alerts

Observe the configured Azure monitor rules for active alerts.

Query Work Items

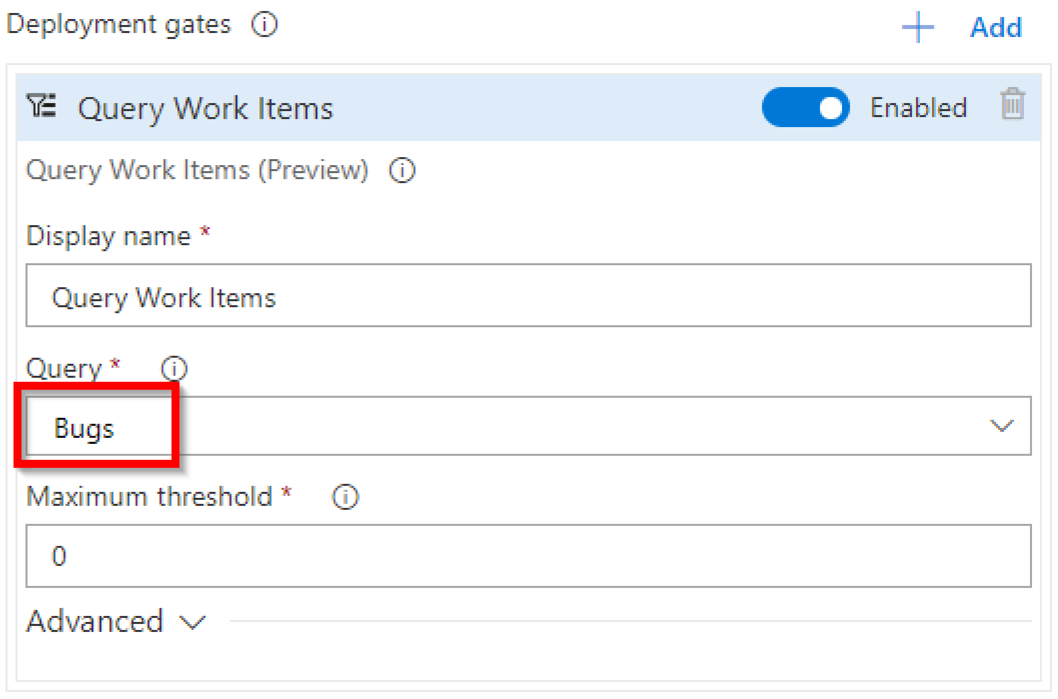
2

Executes a work item query and checks for the number of items returned.

Deployment queue settings

Define behavior when multiple releases are queued for deployment

6. Select **Bugs** under **Shared Queries** in the Query field. As the maximum threshold is set to "0", if this query returns any active bug work Item, the release gate will fail.


Deployment gates

Add

Query Work Items — Enabled

Query Work Items (Preview)

Display name *

Query Work Items

Query *

Bugs

Maximum threshold *

0

Advanced

7. Set the evaluation options.

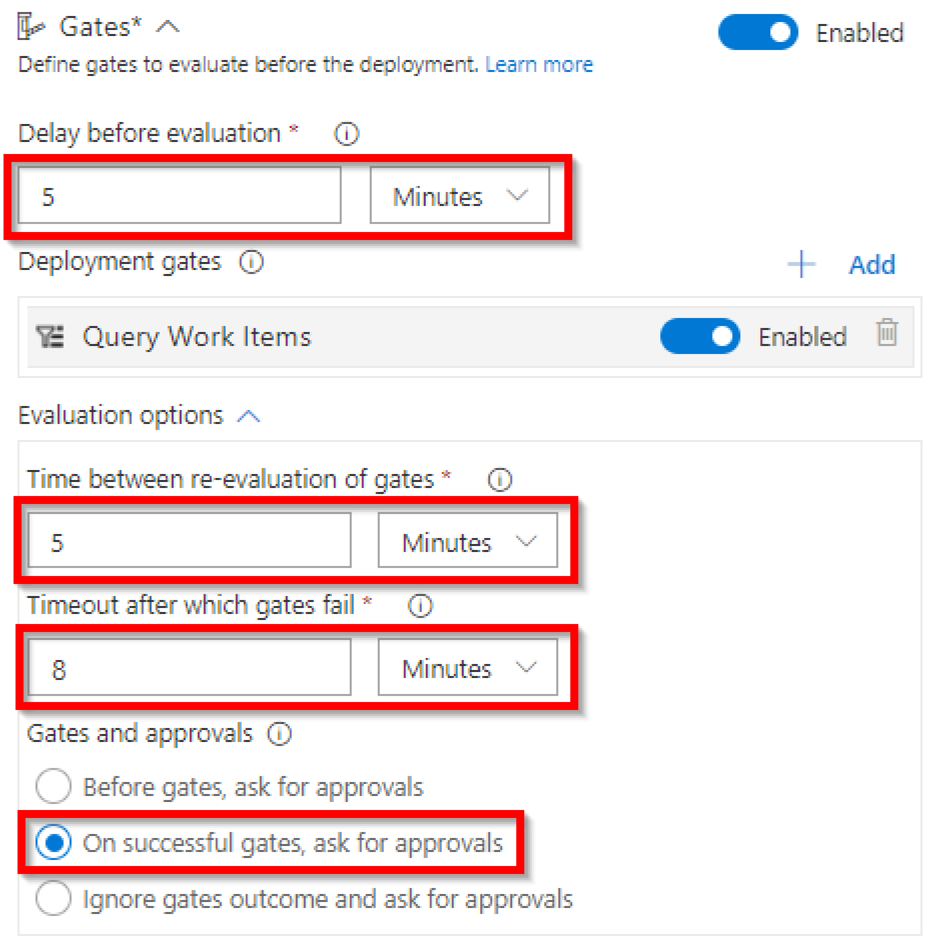
Delay before evaluation: Time before the added gates are evaluated for the first time. If no gates are added, then the deployments wait for the duration before proceeding. To allow gate functions to initialize and stabilize (it may take some time for it to begin returning accurate results), we configure a delay before the results are evaluated and used to determine if the deployment should be approved or rejected.

Time between re-evaluation of gates: The time interval between each evaluation of all the gates. At each sampling interval, new requests are sent concurrently to each gate for fresh results. The sampling interval must be greater than the longest typical response time of any configured gate to allow time for all responses to be received.

Timeout after which gates fail: The maximum evaluation period for all gates. The deployment will be rejected if the timeout is reached before all gates succeed during the same sampling interval. The minimum value we can specify for timeout is 6 minutes and 5 minutes for the sampling interval.

For this demo purpose, set **Delay before evaluation** as *5 minutes* (so that you can see the results reasonably quick), **Time between re-evaluation of gates** as *5 minutes* (sampling interval) and **Timeout after which gates fail** as *8 minutes* but in reality these durations might be in hours. When the release is triggered, gate will validate the samples at *0th and 5th minutes*. If the result is "**Pass**", notification will be sent for approval. If the result is "**Fail**", the release will time-out after *8th minute*.

Select **On successful gates, ask for approvals** radio button.


Gates* Enabled

Define gates to evaluate before the deployment. Learn more

Delay before evaluation *: 5 Minutes

Deployment gates + Add

Query Work Items Enabled

Evaluation options

Time between re-evaluation of gates *: 5 Minutes

Timeout after which gates fail *: 8 Minutes

Gates and approvals

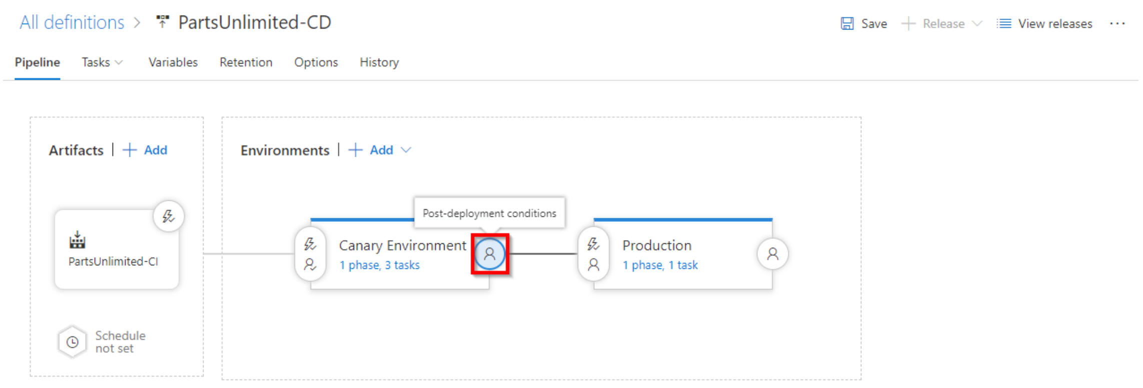
( ) Before gates, ask for approvals

[✓ On successful gates, ask for approvals]

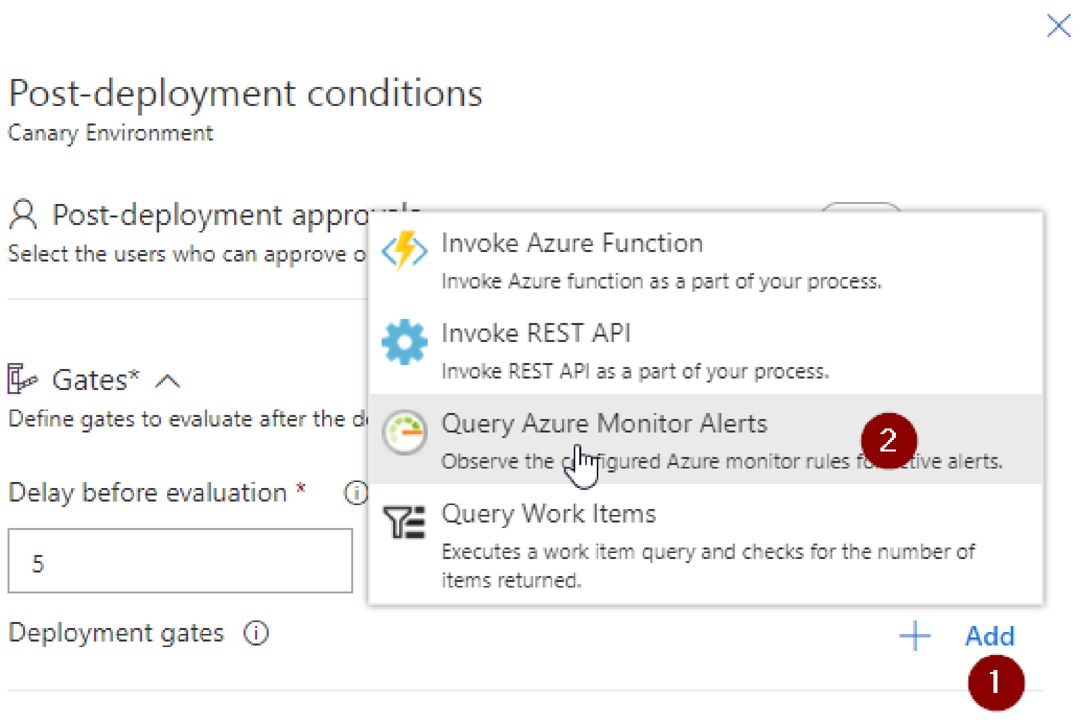
( ) Ignore gates outcome and ask for approvals

Enabling Post-deployment Gate.

1. Click on **Post-deployment conditions**



2. Enable **Gates** and Add **Query Azure Monitor Alerts** to the gate.


Post-deployment conditions
Canary Environment

Post-deployment appro
Select the users who can approve o

Gates*
Define gates to evaluate after the d

Delay before evaluation *
5

Deployment gates

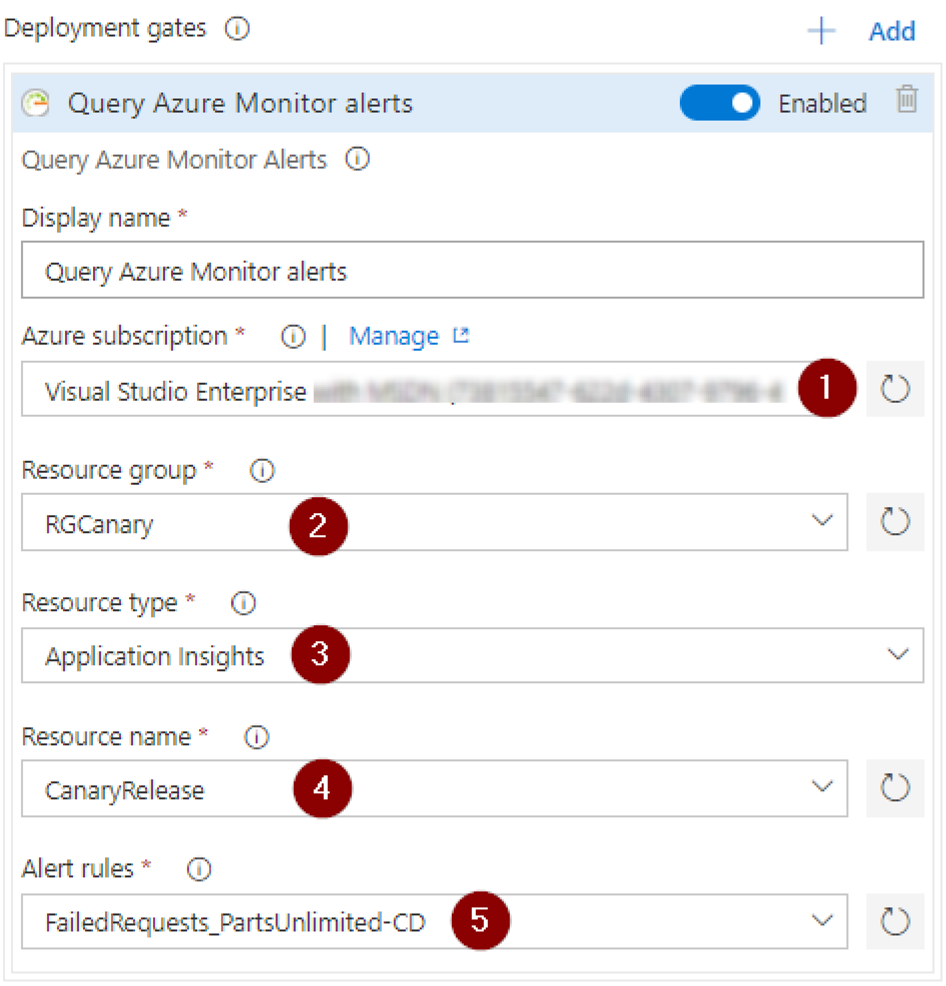
Invoke Azure Function
Invoke Azure function as a part of your process.

Invoke REST API
Invoke REST API as a part of your process.

Query Azure Monitor Alerts
Observe the configured Azure monitor rules for active alerts.

Query Work Items
Executes a work item query and checks for the number of items returned.

3. Update the details from the dropdown.


Deployment gates

Query Azure Monitor alerts — Enabled

Query Azure Monitor Alerts

Display name *: Query Azure Monitor alerts

Azure subscription *: Visual Studio Enterprise [redacted]

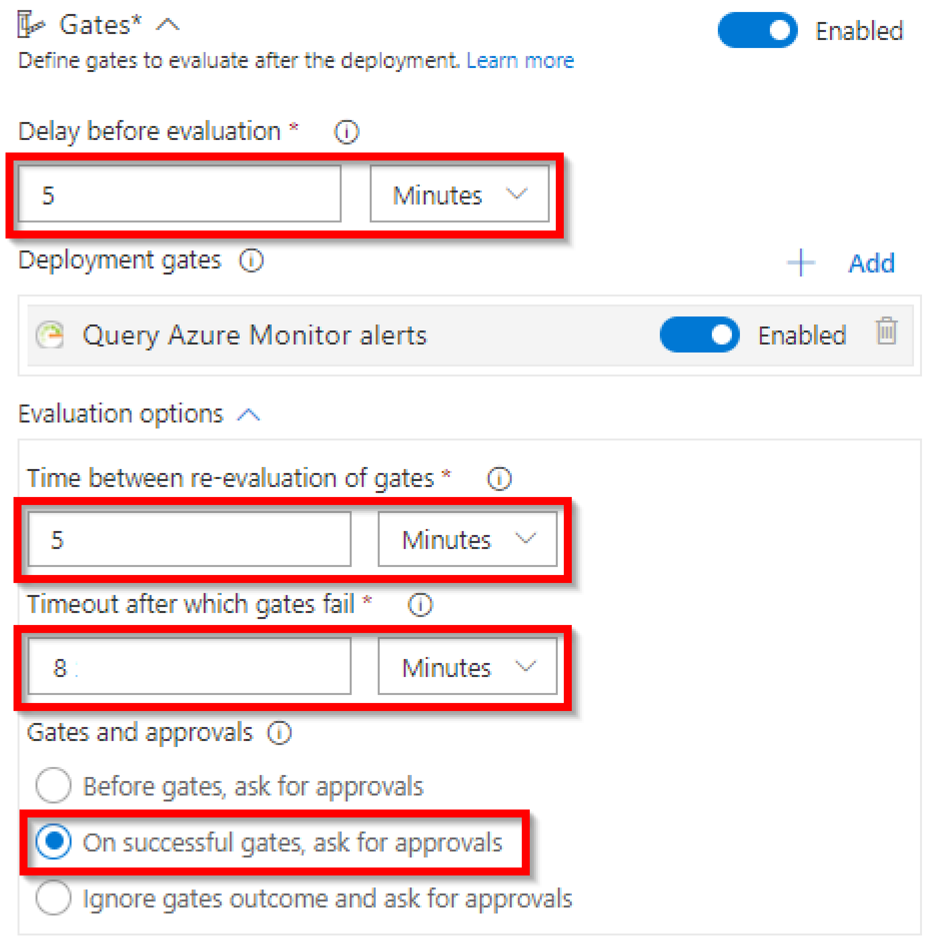
Resource group *: RGCanary

Resource type *: Application Insights

Resource name *: CanaryRelease

Alert rules *: FailedRequests_PartsUnlimited-CD

4. Expand the **Evaluation options** and specify the *delay*, *sampling interval* and the *timeout*. Select **On successful gates, ask for approvals** radio button.


Gates* — Enabled

Define gates to evaluate after the deployment. Learn more

Delay before evaluation *: 5 Minutes

Deployment gates — Add

Query Azure Monitor alerts — Enabled

Evaluation options

Time between re-evaluation of gates *: 5 Minutes

Timeout after which gates fail *: 8 Minutes

Gates and approvals

Before gates, ask for approvals [✓ On successful gates, ask for approvals] Ignore gates outcome and ask for approvals

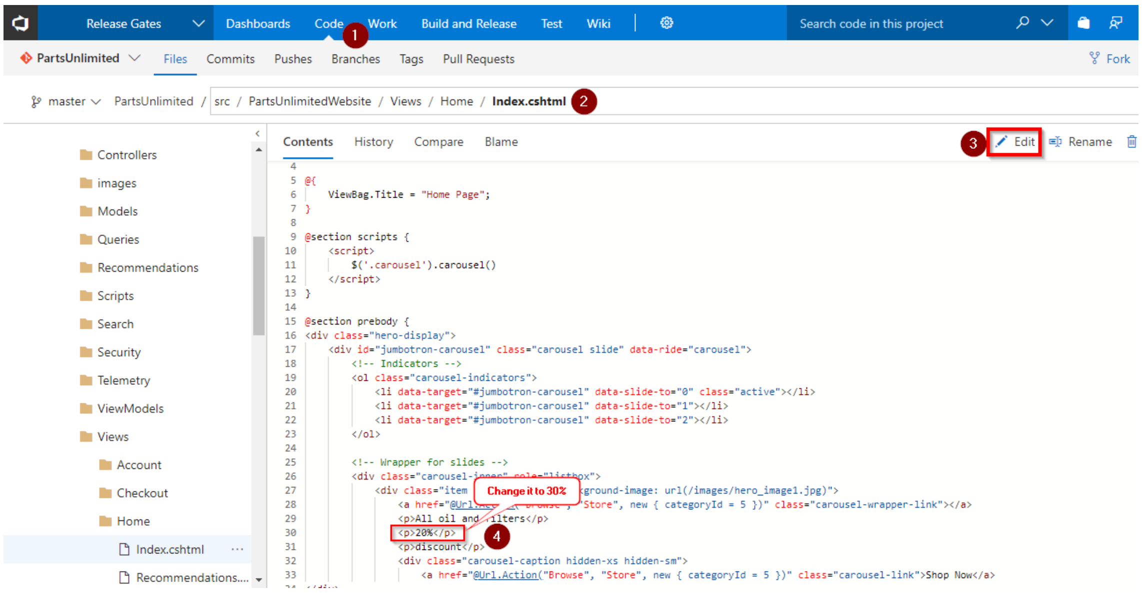
The sampling interval and timeout work together so that the gates will call their functions at suitable intervals and reject the deployment if they don't succeed during the same sampling interval within the timeout period.

5. Click **Save** to save the changes.

Exercise 3: Update and deploy application after adding release gates

In this exercise, you will make a small code change in the application and commit to the repository which in-turn triggers build and release.

1. Go to **Repos** and click *Files*. Navigate to the path *"src/PartsUnlimitedWebsite/Views/Home/Index.cshtml"* and modify the content to ***"30%"***from ***"20%"*** in **line 30**.



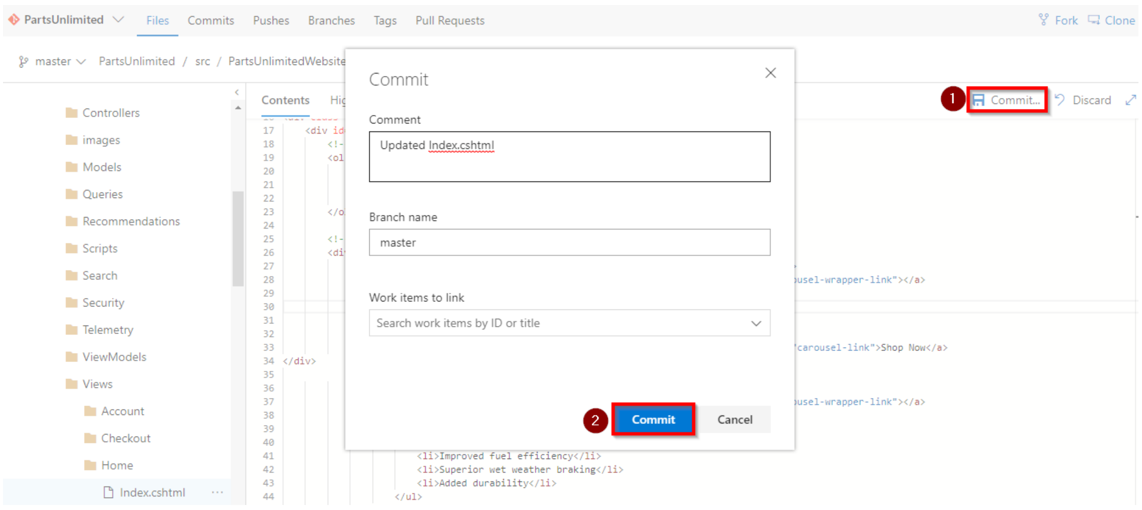
```
@{
    ViewBag.Title = "Home Page";
}

@section scripts {
    <script>
        $('.carousel').carousel()
    </script>
}

@section prebody {
<div class="hero-display">
    <div id="jumbotron-carousel" class="carousel slide" data-ride="carousel">
        <!-- Indicators -->
        <ol class="carousel-indicators">
            <li data-target="#jumbotron-carousel" data-slide-to="0" class="active"></li>
            <li data-target="#jumbotron-carousel" data-slide-to="1"></li>
            <li data-target="#jumbotron-carousel" data-slide-to="2"></li>
        </ol>

        <!-- Wrapper for slides -->
        <div class="carousel-inner" role="listbox">
            <div class="item          ground-image: url(/images/hero_image1.jpg)">
                <a href="@Url.Ac             "Store", new { categoryId = 5 })" class="carousel-wrapper-link"></a>
                <p>All oil and  ilters</p>
                <p>20%</p>
                <p>discount</p>
                <div class="carousel-caption hidden-xs hidden-sm">
                    <a href="@Url.Action("Browse", "Store", new { categoryId = 5 })" class="carousel-link">Shop Now</a>
```

2. After the modification, **Commit** the changes.



Commit

Comment

Updated Index.cshtml

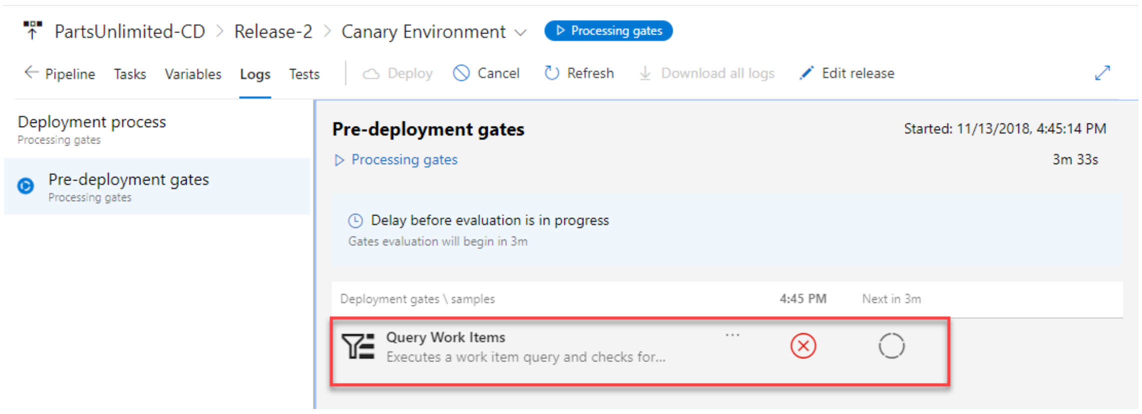
Branch name

master

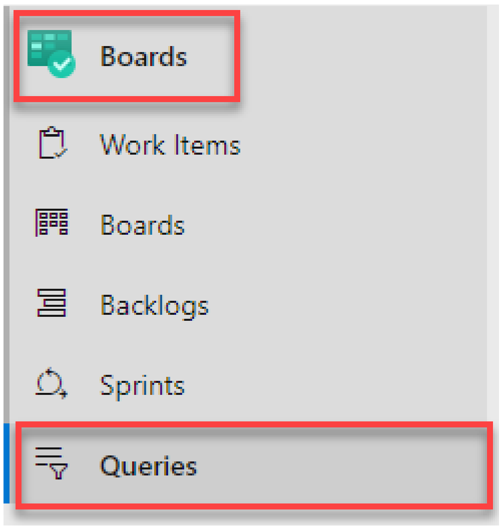
Work items to link

Search work items by ID or title

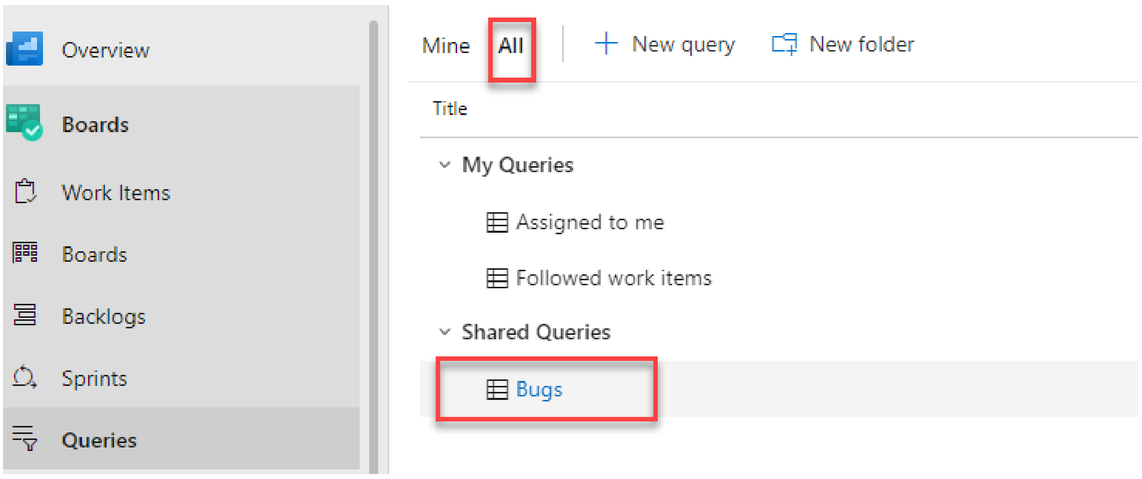
3. The build will automatically trigger as we have *Continuous Integration (CI)* trigger type enabled in the build pipeline. Once the build succeeds, navigate to the **Releases** tab. You will notice the release have been triggered after the successful build.
4. Go to release **Logs** to see the progress. You will see **Query Work Items** gate have failed in a delay before evaluation, which indicates there are active bugs. These bugs should be closed in order to proceed further. Next sampling time will be after 5 minutes.


PartsUnlimited-CD > Release-2 > Canary Environment
Processing gates
Pipeline Tasks Variables Logs Tests
Deploy Cancel Refresh Download all logs Edit release
Deployment process
Processing gates
Pre-deployment gates
Processing gates
Pre-deployment gates
Started: 11/13/2018, 4:45:14 PM
Processing gates
3m 33s
Delay before evaluation is in progress
Gates evaluation will begin in 3m
Deployment gates \ samples
4:45 PM
Next in 3m
Query Work Items
Executes a work item query and checks for...

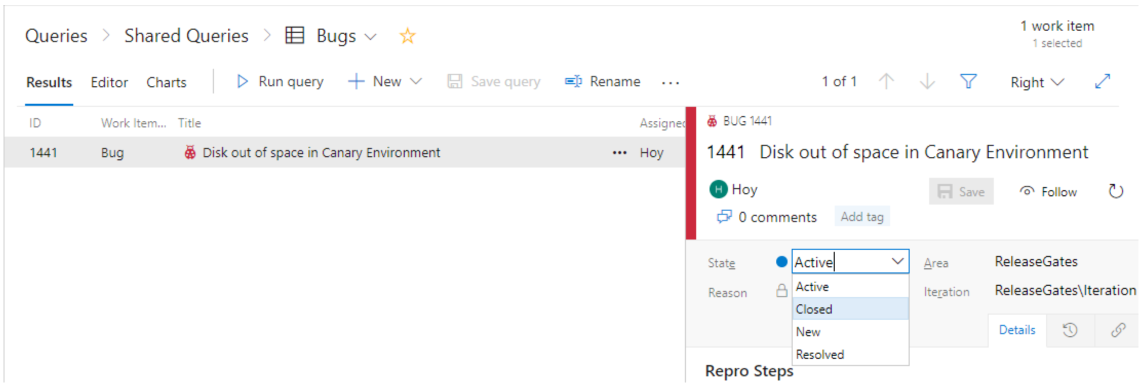
5. Navigate to **Queries** under **Boards** tab.



6. Select **Bugs** under **Shared Queries**.



7. You will see a bug with title “**Disk out of space in Canary Environment**” in **New** State. Assuming that Infrastructure team has fixed the disk space issue, change the state to **Closed** and **Save** it.


Queries > Shared Queries > Bugs

Results Editor Charts | Run query | New | Save query | Rename | 1 of 1 | Right

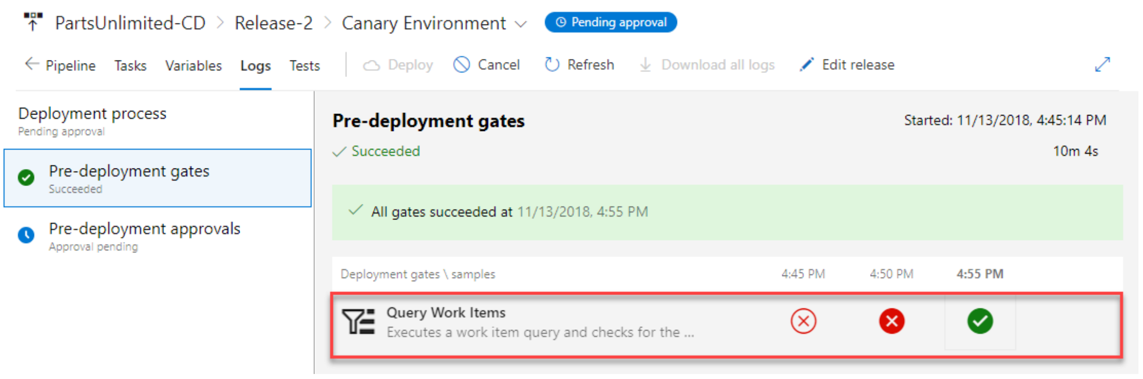
1 work item
1 selected

ID	Work Item...	Title	Assigned
1441	Bug	Disk out of space in Canary Environment	Hoy

BUG 1441
1441 Disk out of space in Canary Environment
Hoy | Save | Follow
0 comments | Add tag

State: Active (Active, Closed, New, Resolved)
Reason
Area: ReleaseGates
Iteration: ReleaseGates\Iteration
Details
Repro Steps

8. Go back to release logs. You will see the evaluation has passed.


PartsUnlimited-CD > Release-2 > Canary Environment | Pending approval

Pipeline Tasks Variables Logs Tests | Deploy | Cancel | Refresh | Download all logs | Edit release

Deployment process
Pending approval

Pre-deployment gates
Succeeded

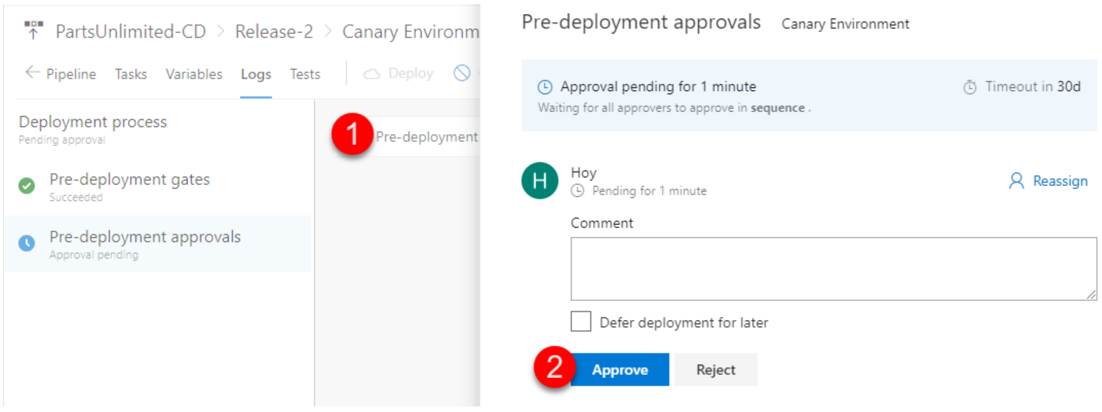
Pre-deployment approvals
Approval pending

Pre-deployment gates
Succeeded
Started: 11/13/2018, 4:45:14 PM
10m 4s

All gates succeeded at 11/13/2018, 4:55 PM

Deployment gates \ samples	4:45 PM	4:50 PM	4:55 PM
Query Work Items Executes a work item query and checks for the ...	✗	✗	✓

9. When the evaluation is successful, you will see the request for pre-deployment approval. Click on **Approve** to deploy in Canary environment


PartsUnlimited-CD > Release-2 > Canary Environm

Pipeline Tasks Variables Logs Tests | Deploy

Deployment process
Pending approval

Pre-deployment gates
Succeeded

Pre-deployment approvals
Approval pending

1 Pre-deployment

Pre-deployment approvals Canary Environment

Approval pending for 1 minute
Waiting for all approvers to approve in **sequence** .
Timeout in 30d

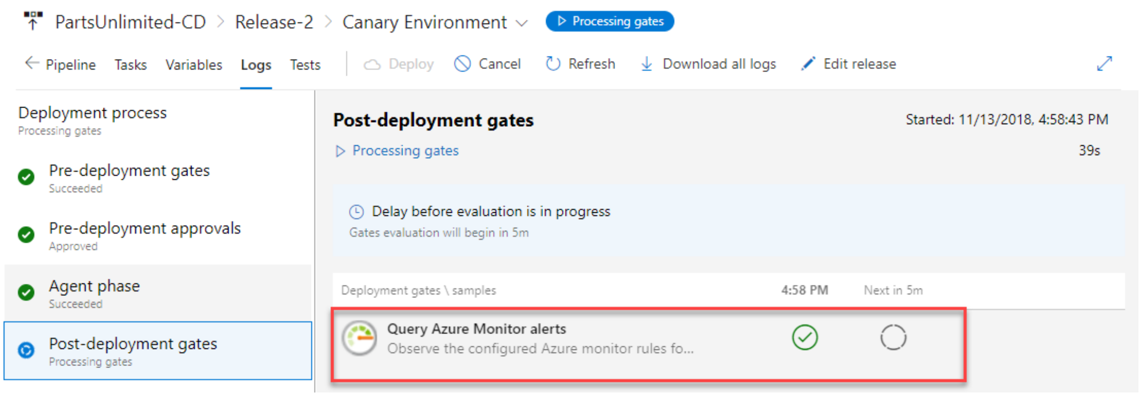
Hoy
Pending for 1 minute
Reassign

Comment

Defer deployment for later

2 Approve Reject

10. Once the deployment to Canary environment is successful, we will see the post-deployment gates in action which will start monitoring the application for any exceptions.



PartsUnlimited-CD > Release-2 > Canary Environment ⌄ ▷ Processing gates

Pipeline | Tasks | Variables | Logs | Tests | Deploy | Cancel | Refresh | Download all logs | Edit release

Deployment process
Processing gates

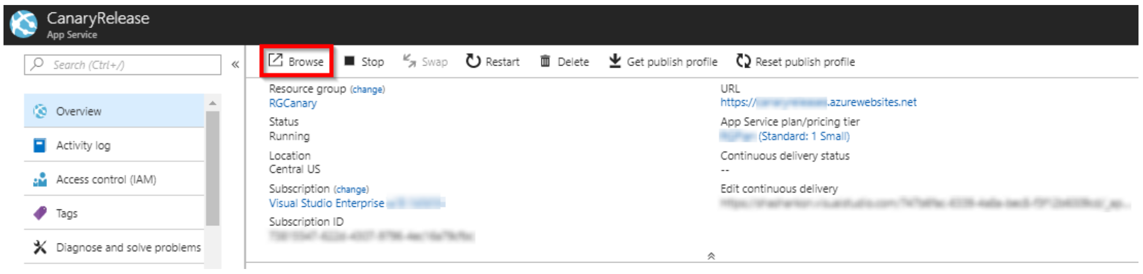
- ✓ Pre-deployment gates — Succeeded
- ✓ Pre-deployment approvals — Approved
- ✓ Agent phase — Succeeded
- Post-deployment gates — Processing gates

Post-deployment gates Started: 11/13/2018, 4:58:43 PM
▷ Processing gates 39s

Delay before evaluation is in progress
Gates evaluation will begin in 5m

Deployment gates \ samples	4:58 PM	Next in 5m
Query Azure Monitor alerts Observe the configured Azure monitor rules fo...	✓	○

11. Now, quickly verify the application. Go to *CanaryRelease* Azure Web App in Azure Portal and click on **Browse**.



CanaryRelease
App Service

Search (Ctrl+/)

Overview | Activity log | Access control (IAM) | Tags | Diagnose and solve problems

Browse | Stop | Swap | Restart | Delete | Get publish profile | Reset publish profile

Resource group (change)
RGCanary

Status
Running

Location
Central US

Subscription (change)
Visual Studio Enterprise [redacted]

Subscription ID
[redacted]

URL
https://[redacted].azurewebsites.net

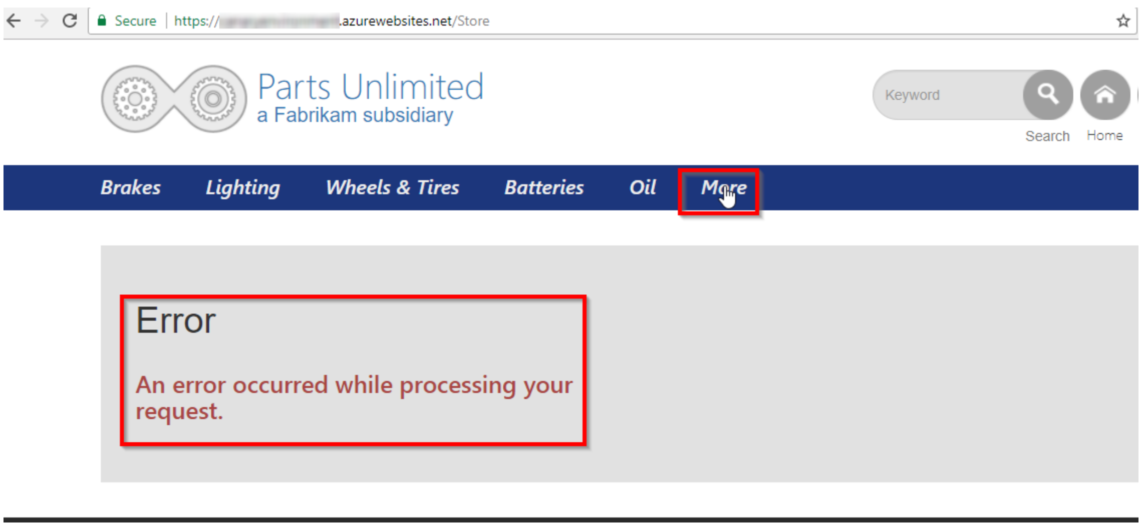
App Service plan/pricing tier
[redacted] (Standard: 1 Small)

Continuous delivery status
--

Edit continuous delivery
[redacted]

12. After the application is launched, click on **More**. You will encounter with an error page. Do this couple of times to trigger alerts.

    This error scenario is just for the purpose of the lab and in the real world, analysis of the alert and a resolution like “disabling a feature flag” or “upgrading the infra” would be realistic.

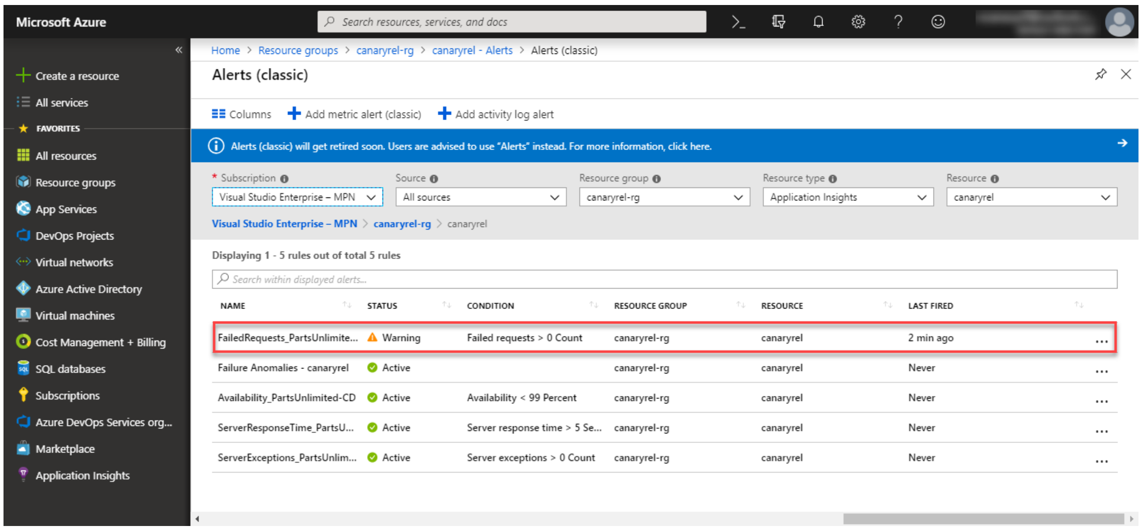


Secure | https://[redacted].azurewebsites.net/Store

Error

An error occurred while processing your request.

13. This exception is monitored by **Application Insights** which will trigger an alert. In Azure Portal, we will be able to see the alert triggered.



Alerts (classic)

Alerts (classic) will get retired soon. Users are advised to use "Alerts" instead. For more information, click here.

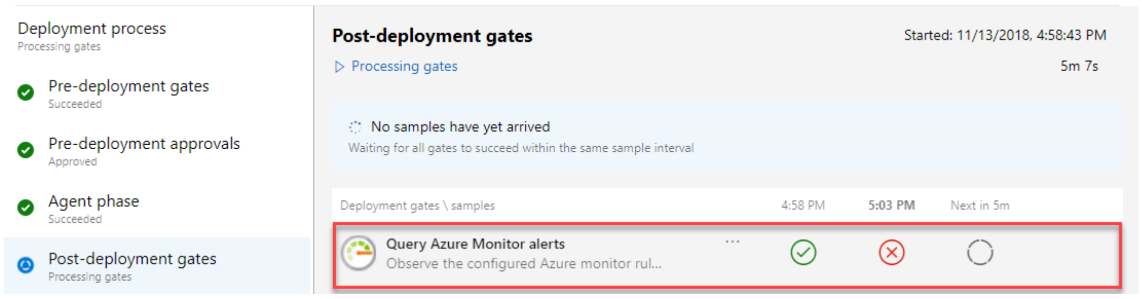
Subscription	Source	Resource group	Resource type	Resource
Visual Studio Enterprise – MPN	All sources	canaryrel-rg	Application Insights	canaryrel

Visual Studio Enterprise – MPN > canaryrel-rg > canaryrel

Displaying 1 - 5 rules out of total 5 rules

NAME	STATUS	CONDITION	RESOURCE GROUP	RESOURCE	LAST FIRED
FailedRequests_PartsUnlimite...	⚠ Warning	Failed requests > 0 Count	canaryrel-rg	canaryrel	2 min ago
Failure Anomalies - canaryrel	✓ Active		canaryrel-rg	canaryrel	Never
Availability_PartsUnlimited-CD	✓ Active	Availability < 99 Percent	canaryrel-rg	canaryrel	Never
ServerResponseTime_PartsU...	✓ Active	Server response time > 5 Se...	canaryrel-rg	canaryrel	Never
ServerExceptions_PartsUnlim...	✓ Active	Server exceptions > 0 Count	canaryrel-rg	canaryrel	Never

14. As there was an alert triggered by the exception, **Query Azure Monitor** gate failed. However, the gate is still under a delay period, you should wait for the next evaluation to proceed.



Deployment process
Processing gates

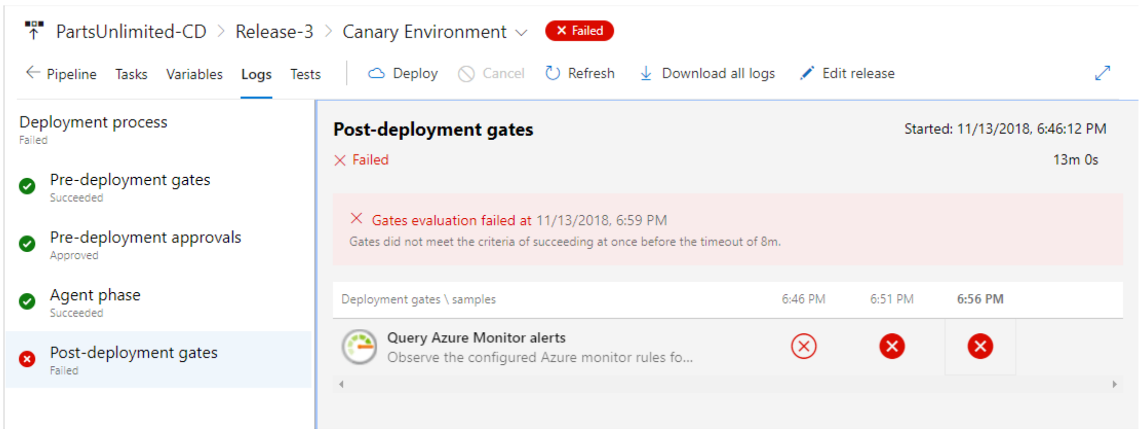
- Pre-deployment gates — Succeeded
- Pre-deployment approvals — Approved
- Agent phase — Succeeded
- Post-deployment gates — Processing gates

Post-deployment gates — Started: 11/13/2018, 4:58:43 PM
Processing gates — 5m 7s

No samples have yet arrived
Waiting for all gates to succeed within the same sample interval

Deployment gates \ samples	4:58 PM	5:03 PM	Next in 5m
Query Azure Monitor alerts Observe the configured Azure monitor rul...	✓	✗	○

15. As the next step, **Query Azure Monitor** gate will block the pipeline and prevents the deployment to **Production** Environment.



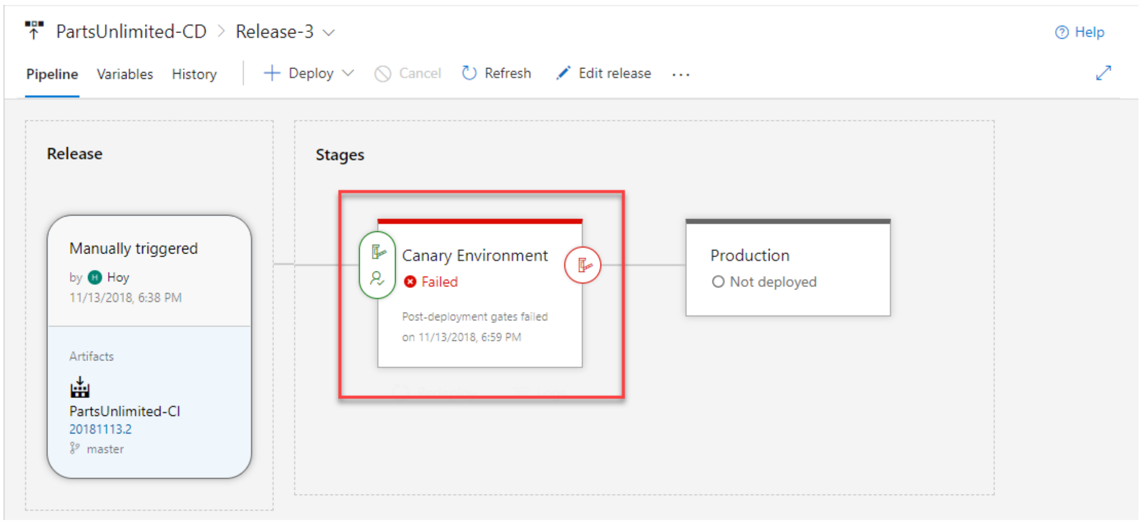
Deployment process
Failed

- Pre-deployment gates — Succeeded
- Pre-deployment approvals — Approved
- Agent phase — Succeeded
- Post-deployment gates — Failed

Post-deployment gates — Started: 11/13/2018, 6:46:12 PM
Failed — 13m 0s

Gates evaluation failed at 11/13/2018, 6:59 PM
Gates did not meet the criteria of succeeding at once before the timeout of 8m.

Deployment gates \ samples	6:46 PM	6:51 PM	6:56 PM
Query Azure Monitor alerts Observe the configured Azure monitor rules fo...	✗	✗	✗



Summary

Gates ensures that the release waits for you to react to the feedback and fix any issues within a timeout period. The gate samples continue to fail and the deployment waits until the issues are fixed. Once the issues are fixed, the next sample from the gates becomes successful and the deployment automatically proceeds.

If a new release is required to fix the issues, then you can cancel the deployment and manually abandon the current release.

So here are release gates, enabling teams to release applications with higher confidence with fewer manual steps. There is now a built-in audit of all the necessary criteria for a deployment being met.